

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A data cache comprising:
a plurality of cache lines, each cache line including a state indicator for indicating anyone of a plurality of states, wherein said plurality of states includes a speculative state to enable keeping track of speculative modification to data in said respective cache line, wherein said speculative state enables a speculative modification to said data in said respective cache line to be made permanent in response to a first operation performed upon reaching a particular instruction boundary during speculative execution of instructions, and wherein said speculative state enables said speculative modification to said data in said respective cache line to be undone in response to a second operation performed upon failing to reach said particular instruction boundary during speculative execution of instructions, wherein said plurality of states includes an invalid state, a dirty state, and a valid state.

2. (Canceled)

3. (Original) The data cache as recited in Claim 1 wherein said first operation is a commit operation.

4. (Original) The data cache as recited in Claim 1 wherein said second operation is a rollback operation.

5. (Original) The data cache as recited in Claim 1 wherein before speculatively modifying a cache line that is in a dirty state, said cache line is cleaned by writing back to memory the dirty data of said cache line.

6. (Previously Presented) A system comprising:
a data cache comprising a plurality of cache lines, each cache line including a state indicator for indicating anyone of a plurality of states, wherein said plurality of states includes a speculative state to enable keeping track of speculative modification to data in said respective cache line, wherein said speculative state enables a speculative modification to said data in said respective cache line to be made permanent in response to a first operation performed upon reaching a particular instruction boundary during speculative execution of instructions, and wherein said speculative state enables said speculative modification to said data in said respective cache line to be undone in response to a second operation performed upon failing to reach said particular instruction boundary during speculative execution of instructions; and
a speculative cache buffer for receiving cache lines which are evicted from said data cache and have one of a second plurality of states.

7. (Original) The system as recited in Claim 6 wherein said plurality of states includes an invalid state, a dirty state, and a valid state, and wherein said second plurality of states includes said speculative state and said dirty state.

8. (Original) The system as recited in Claim 6 wherein said first operation is a commit operation.

9. (Original) The system as recited in Claim 6 wherein said second operation is a rollback operation.

10. (Original) The system as recited in Claim 6 wherein before speculatively modifying a cache line that is in a dirty state in said data cache, said cache line is evicted to said speculative cache buffer.

11. (Original) The system as recited in Claim 6 wherein each cache line in said speculative cache buffer includes a second state indicator for indicating anyone of a plurality of second states, said plurality of second states includes a particular state to indicate that said data cache has evicted said respective cache line that has dirty data in response to a speculative modification operation to said respective cache line in said data cache.

12. (Original) The system as recited in Claim 6 wherein said speculative cache buffer is fully associative.

13. (Currently Amended) A system comprising:

a data cache comprising a plurality of cache lines, each cache line including a state indicator for indicating anyone of a plurality of states, wherein said plurality of states includes a speculative state to enable keeping track of speculative modification to data in said respective cache line, wherein said speculative state enables a speculative modification to said data in said respective cache line to be made permanent in response to a first operation performed upon reaching a particular instruction boundary during speculative execution of instructions, and wherein said speculative state enables said speculative modification to said data in said respective cache line to be undone in response to a second operation performed upon failing to reach said particular instruction boundary during speculative execution of instructions, wherein before speculatively modifying a cache line that is in a dirty state, said cache line is cleaned by writing back to memory the dirty data of said cache line; and

a processor operative to perform a speculative store operation to said data cache.

14. (Original) The system as recited in Claim 13 wherein said first operation is a commit operation.

15. (Original) The system as recited in Claim 13 wherein said second operation is a rollback operation.

16. (Canceled)

17. (Original) The system as recited in Claim 13 further comprising a speculative cache buffer for receiving cache lines which are evicted from said data cache and have one of a second plurality of states.

18. (Original) The system as recited in Claim 17 wherein said plurality of states includes an invalid state, a dirty state, and a valid state, and wherein said second plurality of states includes said speculative state and said dirty state.

19. (Original) The system as recited in Claim 17 wherein before speculatively modifying a cache line that is in a dirty state in said data cache, said cache line is evicted to said speculative cache buffer.

20. (Original) The system as recited in Claim 17 wherein each cache line in said speculative cache buffer includes a second state indicator for indicating anyone of a plurality of second states, said plurality of second states includes a particular state to indicate that said data cache has evicted said respective cache line that has dirty data in response to a speculative modification operation to said respective cache line in said data cache.

21. (Original) The system as recited in Claim 17 wherein said speculative cache buffer is fully associative.

22. (Currently Amended) A method of managing speculative data modifications in a data cache, said method comprising:

in response to a speculative modification to data in a cache line, setting a state indicator of said cache line to a speculative state;

if said speculative modification to said data is desired to be made permanent, changing said state indicator to a first state in response to a first operation performed upon reaching a particular instruction boundary during speculative execution of instructions, wherein said first state is a dirty state; and

if said speculative modification to said data is desired to be undone, changing said state indicator of said cache line to a second state in response to a second operation performed upon failing to reach said particular instruction boundary during speculative execution of instructions.

23. (Canceled)

24. (Original) The method as recited in Claim 22 wherein said second state is a invalid state.

25. (Original) The method as recited in Claim 22 wherein said first operation is a commit operation.

26. (Original) The method as recited in Claim 22 wherein said second operation is a rollback operation.

27. (Original) The method as recited in Claim 22 wherein before speculatively modifying a cache line that is in a dirty state, said cache line is cleaned by writing back to memory a dirty data of said cache line.

28. (Currently Amended) A method of managing speculative data modifications in a speculative cache buffer, said method comprising:

receiving from a data cache a cache line in a speculative state and having speculative modification to the data within said cache line;

setting a state indicator of said cache line to a speculative state in said speculative cache buffer;

if said speculative modification to said data is desired to be made permanent, changing said state indicator to a first state in response to a first operation performed upon reaching a particular instruction boundary during speculative execution of instructions, wherein said first state is a dirty state; and

if said speculative modification to said data is desired to be undone, changing said state indicator of said cache line to a second state in response to a second operation performed upon failing to reach said particular instruction boundary during speculative execution of instructions.

29. (Canceled)

30. (Original) The method as recited in Claim 28 wherein said second state is an invalid state.

31. (Original) The method as recited in Claim 28 wherein said first operation is a commit operation.

32. (Original) The method as recited in Claim 28 wherein said second operation is a rollback operation.

33. (Previously Presented) A method of managing speculative data modifications in a speculative cache buffer, said method comprising:

receiving from a data cache a cache line in response to a speculative modification of data in said cache line within said data cache;

setting a state indicator of said cache line to a commit-kill state in said speculative cache buffer;

if said speculative modification to said data is desired to be made permanent, changing said state indicator to a first state in response to a first operation performed upon reaching a particular instruction boundary during speculative execution of instructions; and

if said speculative modification to said data is desired to be undone, changing said state indicator of said cache line to a second state in response to a second operation performed upon failing to reach said particular instruction boundary during speculative execution of instructions.

34. (Original) The method as recited in Claim 33 wherein said first state is an invalid state.

35. (Original) The method as recited in Claim 33 wherein said second state is a dirty state.

36. (Original) The method as recited in Claim 33 wherein said first operation is a commit operation.

37. (Original) The method as recited in Claim 33 wherein said second operation is a rollback operation.